Generating Realistic Multi-class Biosignals with BioSGAN: A Transformer-based Label-guided Generative Adversarial Network

Xiaomin Li, Anne Hee Hiong Ngu, Vangelis Metsis

2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE) | 979-8-3503-2759-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/CSCE60160.2023.00008

***Abstract*—Time series data are commonly used in biomedical applications, but such datasets are often small, expensive to collect, and may involve privacy issues that restrict large-scale deep learning models. Data augmentation techniques for time series data are limited by the need to maintain signal properties, but Generative Adversarial Networks (GANs) offer a promising approach for expanding datasets. This paper presents BioSGAN, a transformer-based label-guided GAN model capable of generating multi-class, class-specific synthetic time-series sequences of arbitrary length. Our proposed model architecture and design strategies produce synthetic sequences that are almost indistinguishable from real signals, enabling data augmentation. To evaluate the quality of the generated data, we propose a wavelet coherence metric that compares the similarity of real and synthetic signals. Our results show that BioSGAN outperforms existing state-of-the-art time-series GAN models, as demonstrated by qualitative visualizations using PCA and t-SNE, as well as quantitative comparisons of the discriminative and predictive power of the synthetic data.**

BioSGAN source code: https://github.com/imics-lab/BioSGAN

***Index Terms*—Deep Learning, Generative adversarial network, Time-series data generation, Biosignal, Transformer.**

I. Introduction

The availability of sufficient training data is a critical challenge when analyzing time series with deep learning models. While deep learning models have shown remarkable success in computer vision and natural language processing tasks, which often involve image and text data readily available on the internet, this is not the case for time-series data that results from physical or biological processes, especially those involving human subjects. The collection, annotation, and interpretation of such data, such as Electroencephalography (EEG) data used to record brain function, require considerable resources, including the expertise of neuroscience professionals. Moreover, data sharing in medical fields is subject to stringent privacy constraints, further limiting the availability of large medical datasets.

Xiaomin Li, Ph.D. candidate, e-mail:x_l30@txsatate.edu
Anne Hee Hiong Ngu, Professor, e-mail: angu@txstate.edu
Vangelis Metsis, Corresponding Author, Associate Professor, e-mail: vmetsis@txstate.edu
Xiaomin Li and Vangelis Metsis are members of the Intelligent Multimodal Computing and Sensing (IMICS) Lab. All authors are from the department of computer science, Texas State University. Address: 601 University Drive, San Marcos, TX, USA, 78666.

We acknowledge this paper supported by the National Science Foundation under the NSF-SCH grant (2123749)

Despite the success of deep learning models in many areas, their efficacy hinges on the availability of sufficiently large and diverse training datasets. When the size of the dataset is insufficient, deep learning models tend to over-fit the data, leading to poor generalization performance. Researchers are often forced to train shallower deep learning models that cannot fully capture the complexity of the underlying problem. For example, [1] proposed two CNN models with only two and four convolutional layers to classify EEG signals, while [2] leveraged transfer learning and trained an EEG feature extractor with a vast amount of signals but could only use a CNN model with six convolutional layers. Compared to deep learning models used in computer vision tasks, such models have much simpler designs with fewer layers.

Training deep learning models on multi-class, imbalanced datasets can lead to biased model performance in different classes, as shown by [3]. However, data imbalance is a natural phenomenon that exists in many multi-class biomedical datasets, owing to differences in the probabilities of different events occurring. For example, abnormal heartbeat signals resulting from diseases are much rarer and more challenging to collect than normal heartbeat signals, yet they are crucial for disease diagnosis. Researchers have developed various techniques to mitigate data imbalance and produce bias-free deep learning models, including undersampling the majority classes or oversampling the minority classes, assigning higher weights to the minority classes, and using alternative evaluation metrics. These methods can effectively eliminate imbalances to some extent, but they can also lead to information loss, which may ultimately degrade the learning capabilities of deep learning models. [4] conducted a thorough survey of traditional and deep learning-based methods to address data imbalance in multi-class biomedical datasets.

The use of Generative Adversarial Networks (GANs) has gained momentum in the deep learning research field since its introduction in 2014 ([5]). GANs have been successfully applied to generate and manipulate data in computer vision and natural language processing (NLP) domains, including high-quality image generation ([6]), style transfer ([7]), and text-to-image synthesis ([8]), among others. The ability of GAN models to generate new synthetic data from limited amounts of data provides a means of addressing data scarcity issues by adding reproducible synthetic data. Recently, there has been a growing interest in using GANs for time series and sequential data generation. [9] provided a comprehensive overview of

979-8-3503-2759-5/23/$31.00 ©2023 IEEE
DOI 10.1109/CSCE60160.2023.00008

GAN implementations on time-series data and highlighted the advantages of using GAN as a time-series data augmentation tool. For example, GAN can be used to solve data shortage issues by augmenting smaller datasets and generating new, previously unseen data. It can also be employed to recover missing or corrupted data and reduce data noise. Furthermore, it can protect data privacy by generating differentially private datasets that do not contain sensitive information from the source datasets. The paper also presented several state-of-the-art GAN models and algorithms for generating time-series data, such as C-RNN-GAN ([10]), RCGAN ([11]), TimeGAN ([12]), and SigCWGAN ([13]), which all use recurrent neural networks (RNNs) as the base architecture of their GAN models. However, RNN-based GAN models face challenges in producing long synthetic sequences that are realistic enough to be useful. This is attributed to the fact that the timesteps of a series are processed sequentially, and more recent timesteps have a greater effect on the generation of the next timesteps, making it difficult for RNNs to establish relationships between distant timesteps within a long sequence.

The transformer architecture, which relies on multiple self-attention layers ([14]), has recently become a prevalent deep learning model architecture. It has been shown to surpass many other popular neural network architectures, such as convolutional neural networks (CNNs) over images and recurrent neural networks (RNNs) over sequential data ([15]–[19]), in classification tasks, and it has even displayed properties of a universal computation engine ([20]). Some works have already tried to utilize the transformer model in GAN model architecture design with the goal of either improving the quality of synthetic data or creating a more efficient training process ([21], [22]) for image and text generation tasks. [21], for the first time, built a pure transformer-based GAN model and verified its performance on multiple image synthesis tasks.

Since the transformer was invented to handle long sequences of text data and does not suffer from a vanishing gradient problem, theoretically, a transformer GAN model should perform better than RNN-based models on time-series data. [23] introduced a transformer-based GAN model (TTS-GAN) to generate synthetic time-series data. In that approach, a separate GAN model is trained for each class of the dataset. A disadvantage of TTS-GAN is that if, for some of the classes we only have very few training instances, it is difficult to train a GAN model to generate realistic sequences for those classes.

In this paper, we propose a transformer-based *label guided GAN* for biosignal generation, which we call BioSGAN. BioSGAN is trained on data from all classes at the same time, and we can control which class to generate data for by priming it with the right input. The benefit is that BioSGAN can take advantage of transfer learning effects between classes. Essentially, instead of training a separate model for each class, we are training a model on the whole dataset, thus allowing it to learn better low-level feature representations, while high-level features can be fine-tuned simultaneously for each class at the deeper layers of the network. We demonstrate the efficacy of BioSGAN using similarity metrics and experiments showcasing the effect of synthetic data augmentation on classification tasks.

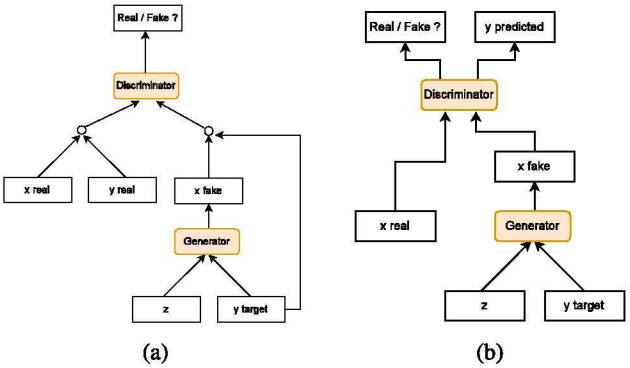


Fig. 1: GAN label embedding strategies. (a) Conventional design, (b) Ours. In both structures, z represents random noise inputs; y represents signal labels; x represents signal data.

Biosignal data are not easily interpretable by humans, and there are no standard metrics used by researchers to compare the similarity of GAN-generated data with real data. To evaluate the similarity between real and synthetic signals, we need a metric that mostly meets the high inter-class variance and low intra-class variance standards. For this reason, we introduce a modified wavelet coherence metric ([24]) to compute the similarity between two sets of signals. Together with dimensionality reduction and visualization methods such as PCA and t-SNE, and some previously proposed metrics such as the discriminative and predictive power of the synthetic data, we are able to demonstrate both qualitatively and quantitatively that our model produces better-quality synthetic signals than other existing models.

Our contributions can be summarized as follows:

- We introduce BioSGAN, a transformer-based label-guided GAN model that can generate multi-class time-series data, and we conduct an empirical study on the best strategy to embed labels on GAN models.
- We modify the wavelet coherence metric to be able to quantitatively compare the similarity between two sets of signals.
- We evaluate our method using multiple qualitative and quantitative metrics and demonstrate its superiority compared to other state-of-the-art GAN models.

II. Background

A. Conditional GANs

In regular GAN models, the generator and discriminator have no control over the data being generated. By adding additional information to these networks, it is possible for them to direct the data generation process. This additional information can include class labels, data from different modalities, and other relevant data. [25] proposed a conditional GAN model training strategy that uses data labels during the training process to generate data belonging to specific categories shortly after the original GAN paper [5]. Recent works in computer vision, such as styleGAN ([26]) and

cycleGAN ([27]), concatenate a different format of the image to the real input data to create new styles of synthetic data.

We assert that conditional GANs can be used to overcome some of the problems encountered in generating time-series data using GANs. In a sensor-collected time-series dataset with multiple categories of signals, regular GANs require training a GAN model on only the type of real signal that corresponds to a particular category of synthetic signal. Therefore, multiple GAN models must be trained to obtain all types of signals for a multi-category dataset. Additionally, based on empirical experiments, we noticed that the lower the amount of data used for training the GAN model, the easier it is for the discriminator to overfit the training data, and the lower the quality of the generated data. Furthermore, available bio-signal time series datasets are often small and have imbalanced numbers of samples for each class. Therefore, training multiple GAN models on each category of data for a small multi-class dataset is neither efficient nor effective. However, currently, no work has been done on using labels as control conditions for GAN models to generate multi-category time-series data.

Conventional label-guided GAN designs that generate different categories of data provide label information for both the generator and discriminator, as shown in Fig.1a. Inspired by the works of [28] and [29], we propose a new strategy for embedding label information into GAN models. Specifically, we only embed labels on the generator and add another classification head on the discriminator. The generator generates multi-category synthetic data, and the discriminator is trained to distinguish the real and synthetic signals and their corresponding categories, as shown in Fig.1b. We tested this idea on many GAN model architectures, and all of them performed similarly to or better than the conventional method. Therefore, we adopted this label embedding idea in our BioSGAN model architecture.

B. Time-series Similarity Analysis

In assessing the quality of GAN-generated images, researchers frequently use questionnaires to collect feedback on the generated images. However, time-series data can be difficult to interpret visually, in contrast to images that can be readily interpreted by the human eye. Researchers often use dimensionality reduction techniques such as PCA and t-SNE as qualitative methods to distinguish major feature distributions of time-series datasets. Nevertheless, we have observed that, even though most state-of-the-art GAN models can generate synthetic data with similar PCA and t-SNE distributions, the synthetic raw data patterns are not necessarily visually similar. As a result, we require a more robust and quantitative technique to evaluate the quality of generated signals.

Various time-series similarity metrics exist, such as the point-to-point distance measure, dynamic time warping, longest common sub-sequence, among others ([30]). However, we are not seeking absolute similarity (or distance) between real and synthetic signals. Similar to how there are different cat images labeled as a cat, a set of time-series data representing

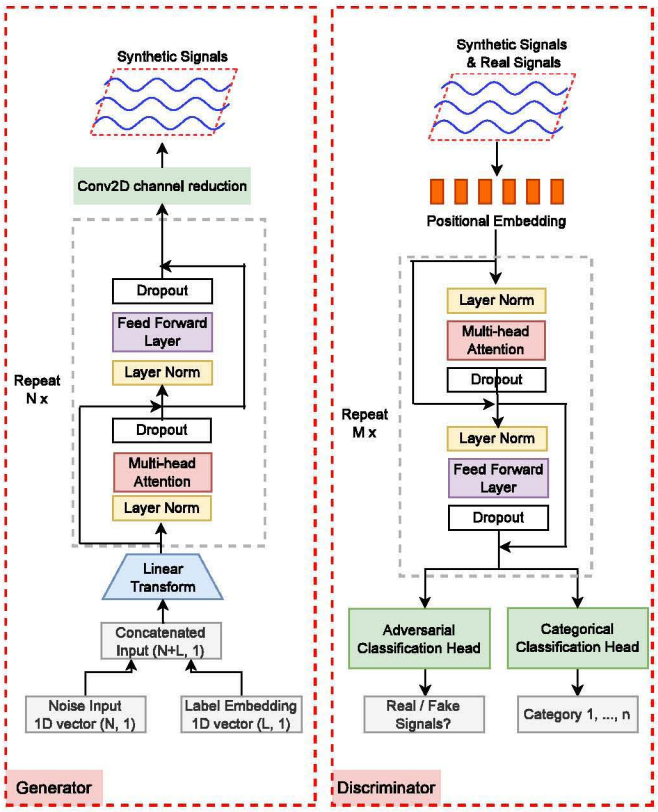


Fig. 2: BioSGAN model architecture.

the same event may also have different signal patterns. We need a metric that can evaluate how similar a set of synthetic signals is to a set of real signals. In their work, [23] proposed two metrics, average cosine similarity and average Jensen-Shannon distance, which require manual feature extraction from raw sequences for comparison. The effectiveness of these metrics depends heavily on the feature extraction methods.

In this paper, we propose a modification to the wavelet coherence metric ([24]) to enable the measurement of similarity between two sets of signals without the need for extra machine learning models or feature extraction. Details of the metric are provided in Section III-D. Our experimental results demonstrate that BioSGAN-generated data display significantly higher wavelet coherence scores than other baseline models. Using this metric to rank state-of-the-art GAN models, our results are consistent with previous works ([12], [13]).

III. Methodology

A. BioSGAN Model Architecture

The BioSGAN model architecture is shown visually in Fig. 2. It contains two main components, a generator and a discriminator. Both of them are built based on the transformer encoder architecture ([14]). An encoder is a composition of two compound blocks. A multi-head self-attention module constructs the first block, and the second block is a feed-forward multi-layer perceptron (MLP) with a GELU activation function. A normalization layer is applied before both of the

two blocks, and the dropout layer is added after each block. Both blocks employ residual connections.

The generator first takes in a 1D vector with N uniformly distributed random numbers $(N, 1)$ within the range (0,1), i.e. $N_i \sim U(0, 1)$. N represents the latent dimension of the synthetic signals, which is a hyperparameter that can be tuned. A target synthetic data label embedding vector $(L, 1)$ is concatenated to the random noise vector to generate a new vector $(N + L, 1)$. This vector is then mapped to a sequence with the same length as the real signals and M hidden dimensions. M is also a hyperparameter that can be changed and is not necessarily equal to the real signal dimensions. This sequence is then input to the transformer encoder blocks. Then encoder outputs are passed through a Conv2D layer to reduce the synthetic data dimensions from M to the real signal dimensions. For example, a synthetic data sequence after the generator transformer encoder layers with a data shape $(hidden\_dimensions, 1, timesteps)$ will be mapped to $(real\_data\_dimensions, 1, timesteps)$. In this way, a random noise vector is transformed into a sequence with the same shape as the real signals.

The discriminator architecture is similar to the ViT model ([15]). The major difference is the ViT model cuts images into small patches and is used for image classification tasks, but in the BioSGAN model, the inputs are multi-dimensional time-series signals. Due to the property of signals that the feature values at a single time point do not have significant meanings, we consider an input signal segment as an entirety and set a patch size to cut input signals into multiple time windows as the ViT model does. And then add positional encoding to each patch. The discriminator D we developed contains two classification heads. One is the adversarial head that will predict whether the input signal is real or fake, D_{adv}, and another is the category head that predicts what category this signal belongs to, D_{cls}. The discriminator outputs can be written as $D : x \rightarrow \{D_{adv}(x), D_{cls}(x)\}$.

B. Updating the Model Parameters

The objective functions to optimize Generator G and Discriminator D can be written as:

$$L_D = -L_{adv} + \lambda L^r_{cls}, \tag{1}$$

$$L_G = L_{adv} + \lambda L^f_{cls}, \tag{2}$$

where L_{adv} is the adversarial loss to determine how well D can distinguish real and fake signals. L_{cls} is the classification loss to determine how well D can classify the input signal to its proper categorical label. λ is a hyper-parameter that controls the relative importance of categorical classification loss and adversarial loss. We use $\lambda = 1$ in all of our experiments.

1) Adversarial loss: To make the generated synthetic signals indistinguishable from real signals, we adopt an adversarial loss

$$L_{adv} = E_x[logD_{adv}(x)] + E_{z,c}[log(1 - D_{adv}(G(z, c)))] \tag{3}$$

where G generates a signal $G(z, c)$ conditioned on both the random noise input z and the target categorical label c, while D tries to distinguish between real and fake signals. During the GAN model training, the generator G tries to minimize the adversarial loss, while the discriminator D tries to maximize it. Therefore, in equation 1, we add a negative symbol in front of the L_{adv}. By minimizing $-L_{adv}$, the discriminator maximizes the L_{adv} loss.

2) Categorical loss: For a random noise vector z in latent dimension and a target categorical label c, our goal is to map z into an output signal, which is properly classified to the target category c. To achieve this goal, we add another classification head on the discriminator D and impose the categorical classification loss when optimizing both D and G. We decompose this objective into two terms, a categorical classification loss of real signals used to optimize D, it can be defined as:

$$L^r_{cls} = E_{x,c'}[-logD_{cls}(c'|x)] \tag{4}$$

where L^r_{cls} is the categorical classification loss on real signals. c' is a real signal categorical label. By minimizing this loss function, D learns to classify a real signal x to its corresponding original category c'.

Another term, categorical classification loss of fake signals, is used to optimize G, which can be defined as:

$$L^f_{cls} = E_{z,c}[-logD_{cls}(c|G(z, c))] \tag{5}$$

where L^f_{cls} is the categorical classification loss on synthetic(fake) signals. c is a target signal categorical label randomly generated. By minimizing this loss function, G learns to generate signals that can be classified as the target category c.

3) Wasserstein loss: To stabilize the training process and generate higher quality signals, we replace equation 3 with Wasserstein GAN objective with gradient penalty ([31], [32]), defined as

$$\begin{aligned} L_{adv} = E_x[D_{adv}(x)] - E_{z,c}[D_{adv}(G(z, c))] \\ -\lambda_{gp} E_{\hat{x}}[(\|\nabla_{\hat{x}} D_{adv}(\hat{x})\|_2 - 1])^2] \end{aligned} \tag{6}$$

where $\hat{x}$ is sampled uniformly along a straight line between a pair of real and synthetic signals. We use $\lambda_{gp} = 10$ for all experiments.

C. Label Embedding Strategies

In the study, we aimed to find the optimal label embedding strategy for a conditional GAN model. Since the quality of time series is difficult to quantify, we used the MNIST digit dataset as a case study, to generate synthetic digit images. To accomplish this, we flattened the digit images into 1D vectors (similar to a time-series sequence) and tested a variety of label embedding methods to generate digit images. We then evaluated the fidelity of the generated images by reshaping the vectors to their original image size and comparing them to the original images.

The first three strategies involved concatenating or adding label embeddings to both the generator and discriminator inputs of linear or convolutional GAN models. The fourth strategy, which performed the best among all strategies, only

concatenated the label embedding to the generator and added a classification head to the discriminator to classify the input categories. From the experimental findings, we discovered that simple concatenation of label embeddings to both the generator and discriminator inputs is a naive way to build a conditional GAN model, while adding label embeddings as in transformer positional encoding is not practical for label embedding.

A detailed explanation of each embedding strategy and sample output results are given in the Appendix.

D. Wavelet Coherence for twos sets of signals

Wavelet coherence ($wcoh$) ([24]) measures the correlation between two signals. A detailed explanation about the efficacy of Wavelet Coherence can be found in the appendix. The $wcoh$ of two time-series vectors x and y is:

$$wcoh = \frac{|S(C_x^*(a,b)C_y(a,b))|^2}{S(|C_x(a,b)|^2) \cdot S(|C_y(a,b)|^2)} \tag{7}$$

where $C_x(a,b)$ and $C_y(a,b)$ denote the continuous wavelet transforms of x and y at scales a and position b. The superscript $*$ is the complex conjugate and S is a smoothing operator in time and scale. Wavelet coherence is useful for analyzing non-stationary[1] signals.

Since the wavelet coherence of two signals, $wcoh$ is a 2D matrix with the shape $Frequencies \times Timesteps$. We compute the sum over the x-axis and mean over the y-axis of $wcoh$ to get a scalar value of two signals' wavelet coherence $wcoh\_s$ to use as a score. If these two signals are multi-dimensional, we compute a mean $wcoh\_s$ over all dimensions. For two sets of signals, A and B, each set has a total number of n signal samples. We compute $wcoh\_s$ of every signal in A to every signal in B and then compute the overall mean $wcoh\_set$ as the average wavelet coherence for two sets of signals. This process can be written as:

$$wcoh\_set = \frac{\sum_{i=1}^{n}(\sum_{j=1}^{n} wcoh\_s(A_i, B_j)/n)}{n} \tag{8}$$

In this way, we compute a scale value to measure the similarity between two sets of signals.

IV. Experiments

A. Datasets

We evaluate our model on two datasets. UniMiB human activity recognition (HAR) dataset ([33]), and MIT-BIH Arrhythmia Database ([34], [35]). A few raw data samples for each dataset are shown in Fig. 4 (left).

For the **UniMiB dataset** ([33]), we select 24 subjects' recordings to train the BioSGAN model. The dataset samples are labeled with nine different human activities. For example, Walking, Running, Jumping, Sitting Down, etc. Every sample has 150 timesteps and three acceleration values (x, y, z) at each timestep.

[1]A signal is said to be non-stationary if its frequency of spectral contents is changing with respect to time.

The **MIT-BIH Arrhythmia dataset** ([34], [35]) contains 48 half-hour ambulatory ECG recordings, obtained from 47 subjects in 5 different heart health conditions. They are Non-Ectopic Beats, Superventrical Ectopic, Ventricular Beats, Unknown, and Fusion Beats. Each sequence is pre-segmented to 187 timesteps with an electric potential measurement, in microvolts (mV), at each timestep.

B. Experimental Setup

The experiments were conducted on a workstation with a 3.40GHz CPU, 377GB RAM, and 2 Nvidia 1080 GPUs. The PyTorch deep learning library ([36]) was used for all experiments. The generator's input vector size for all datasets was set to $(100, 1)$. The generator and discriminator both contained three repeated transformer blocks. The learning rates for the generator and discriminator were set to $1e-4$ and $3e-4$, respectively. An Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and a batch size of 32 for both generator and discriminator were used for all experiments. The value of λ in equations 1 and 2 was set to 1, and the value of λ_{gp} in equation 6 was set to 10. These parameter values were selected based on available computation resources, practical experience, and GAN model training conventions. It is possible that case-by-case parameter tuning may lead to better synthetic data quality, but our GAN model outperforms other state-of-the-art models even with these general setups. The real and synthetic data samples were saved to matrix files, and the Wavelet Coherence scores were computed using MATLAB.

C. BioSGAN Evaluation

We evaluate our proposed model using several qualitative and quantitative metrics and compare its performance with multiple state-of-the-art time-series GAN models.

1) Qualitative analysis: Fig. 4 displays samples of raw data visualization of both real and synthetic signals. Due to page limitations, only a few real and synthetic signals from a few data categories are plotted. The first two rows of plots are synthetic jumping and sitting down data generated from the UniMiB human activity recognition dataset. These two examples were selected from a total of 9 categories. It was observed that the real data samples within a category had significant pattern variability due to being collected from multiple subjects. Surprisingly, the BioSGAN model could successfully capture such different patterns and generate a similar synthetic dataset. The bottom two rows of plots are from the MIT-BIH Arrhythmia ECG dataset. A single BioSGAN model was used to generate five categories of synthetic data. From these plots, it was observed that the BioSGAN model could generate multiple categories of data at once, and all of them were of relatively high quality.

From the plots, however, it was noticed that the synthetic signals were noisier than the real signals, and data quality might be affected by how complex the original real dataset was. Further studies may be warranted to determine the cause of such noise, although a post-processing filter can be applied to the generated signals to reduce it.

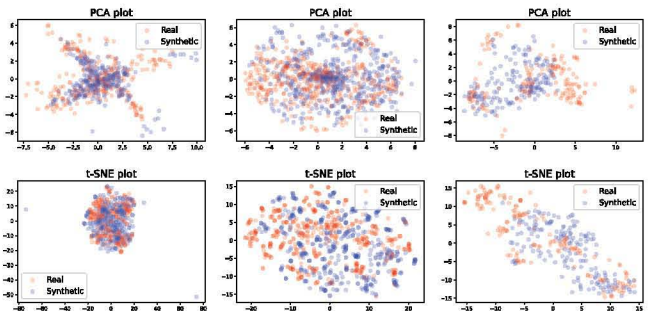


Fig. 3: The PCA and t-SNE visualization for real and synthetic data generated by BioSGAN. (a) Jumping, (b) Running, (c) Sitting Down. Red dots denote real samples, blue dots denote synthetic samples.

To further illustrate the qualitative similarity between the real data and synthetic data, visualizations of data point distributions mapped to two dimensions using Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) were plotted in Fig. 3. In these plots, each dot represents a real or synthetic data sample (sequence) value after dimensionality reduction. Red dots represent real data samples, and blue dots represent synthetic data samples generated by BioSGAN. The top row displays PCA plots from each data category, and the bottom row displays t-SNE plots. Although PCA and t-SNE reduce data dimensions differently, it was observed from these plots that the dots from real data and synthetic data followed a similar distribution per class, indicating that the original high-dimensional real and synthetic data samples shared similarities.

To compare the fidelity of BioSGAN synthetic data with other baseline models, we plot a PCA and t-SNE examples from the MIT-BIH dataset in Fig. 5. The Non-ectopic heartbeat signals are the easiest ones to generate compared with other categorical data we used in this paper. From the plots, we can see that the other baseline models all failed to generate synthetic data that have a very similar distribution to the real data.

More visualization graphs and comparison with other state-of-the-art works can be found in appendix.

2) Quantitative analysis: We employ the Wavelet Coherence score, as introduced in section III-D, as a quantitative measure to demonstrate the similarity between a set of synthetic signals and a set of real signals. A higher score indicates a higher degree of similarity between the two sets. Additionally, we use the *discriminative score* introduced by [12] as an alternative similarity metric. This metric is computed by training a time-series classification model to differentiate between sequences from the original and generated datasets, with a smaller score indicating higher quality synthetic data. Table I presents the metric scores obtained from multiple datasets.

We evaluated the performance of our BioSGAN model against several baseline models using various categories of data with different data properties. Specifically, we chose the

Wavelet Coherence score (the higher the better)				
Models	SittingDown	Jumping	Non-Ectopic	FusionBeats
C-RNN-GAN	41.10	40.29	30.44	25.51
RCWGAN	39.90	38.85	29.72	22.97
TimeGAN	40.45	39.42	31.55	21.98
SigCWGAN	41.60	41.02	31.36	22.87
TTS-GAN	43.92	**47.64**	45.30	55.64
BioSGAN	**45.07**	**47.64**	**47.79**	**58.34**
Discriminative score (the lower the better)				
Models	SittingDown	Jumping	Non-Ectopic	FusionBeats
C-RNN-GAN	0.308	0.304	0.189	0.493
RCWGAN	0.294	0.311	0.483	0.499
TimeGAN	0.261	0.217	0.464	0.312
SigCWGAN	0.310	0.308	0.413	0.491
TTS-GAN	0.294	0.167	**0.107**	0.380
BioSGAN	**0.191**	**0.057**	0.162	**0.261**

TABLE I: Quantitative evaluation of state-of-the-art time-series GANs using the wavelet coherence and discriminative score metrics.

Class	Method	Precision	Recall	F1-score
Non-Ectopic Beats (normal)	Imbalanced	0.94	0.99	0.97
	Augmented	0.97	0.97	0.97
Superventrical Ectopic Beats	Imbalanced	1.00	0.15	0.25
	Augmented	0.67	**0.67**	**0.67**
Ventricular Beats	Imbalanced	0.78	0.72	0.75
	Augmented	0.82	**0.79**	**0.81**
Unknown Beats	Imbalanced	0.80	0.25	0.38
	Augmented	0.39	**0.75**	**0.51**
Fusion Beats	Imbalanced	0.98	0.82	0.89
	Augmented	0.98	**0.89**	**0.89**
Average Imbalanced		0.9	0.59	0.65
Average Augmented		0.76	**0.81**	**0.77**

TABLE II: The classification scores of training a classifier on the original, imbalanced dataset and on an augmented version of the dataset with synthetic data added to the minority classes.

Sitting Down and Jumping sets from the UniMiB dataset and the Non-Ectopic Beats and Fusion Beats from the MIT-BIH Arrhythmia dataset. For each dataset, we trained a BioSGAN model and used it to generate synthetic data with target labels as inputs. We computed quantitative scores based on the similarity between a set of real data and a set of synthetic data from each dataset. The results in Table I indicate that, in most cases, our BioSGAN model outperformed the baseline models. This suggests that our GAN models can generate more realistic synthetic signals than the other GAN models. Notably, the rankings of performance are consistent with those reported in a previous study ([13]).

D. Predictive Power of Synthetic Data

In this experiment, we investigate the effectiveness of using synthetic data to improve the performance of a deep learning model on an imbalanced training set. Specifically, we focus on the MIT-BIH Arrhythmia dataset, which contains five classes of electrocardiogram (ECG) signals with severe class imbalance with ratios of approximately (0.83 : 0.03 : 0.07 : 0.01 : 0.07) corresponding to five classes, Non-Ectopic Beats (normal), Superventrical Ectopic Neats, Ventricular Beats, Unknown Beats, and Fusion Beats. Except for the Non-Ectopic Beats, the other four classes represent some abnormal heartbeats that should be as much as possible to be diagnosed..

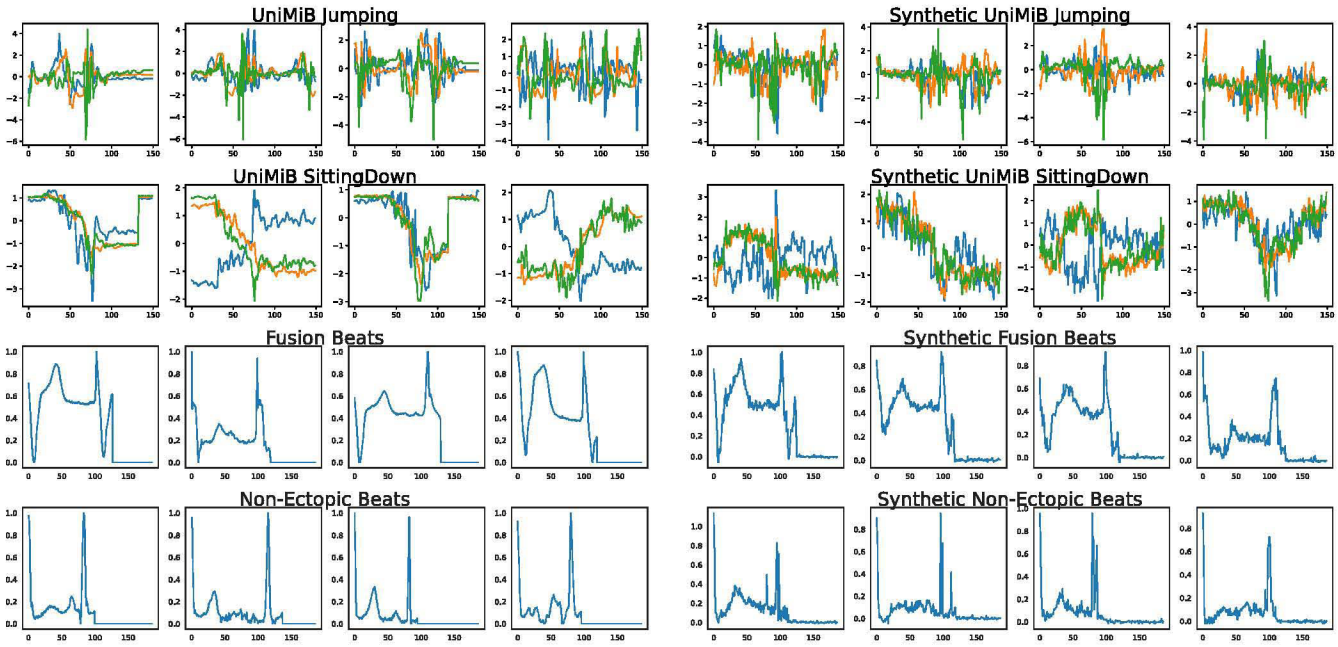
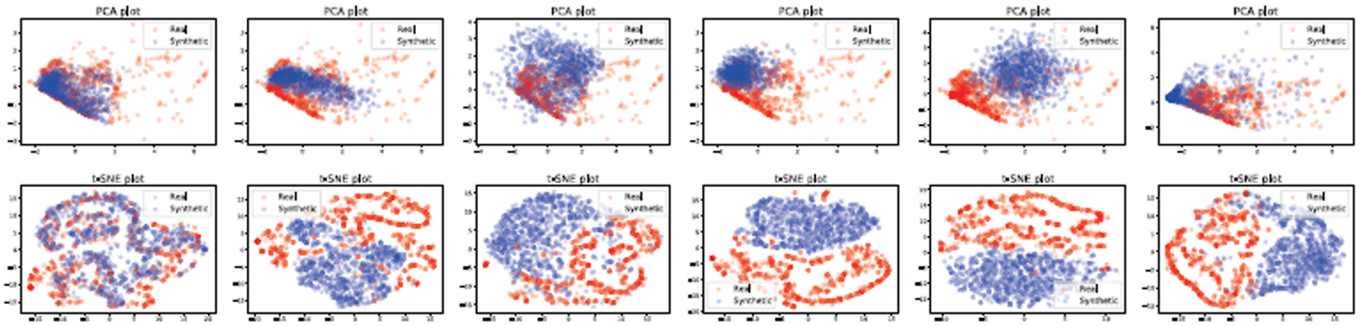


Fig. 4: A visual comparison of real data (left) and their corresponding synthetic data (right) generated by BioSGAN.



(a) BioSGAN (b) TTS-GAN (c) SigCWGAN (d) TimeGAN (e) RCWGAN (f) C-RNN-GAN

Fig. 5: PCA plots (top row) and t-SNE plots (bottom row) showing the distribution of data generated by different GAN models for synthetic Non-Ectopic Beats. Red dots denote real data samples, and blue dots indicate synthetic ones.

We train a convolutional neural network (CNN) model to classify the input ECG signals into one of the five classes. Although the model achieves an average testing accuracy of 93% on the imbalanced dataset, the recall scores for the minority classes, such as Superventricular Ectopic Beats and Unknown Beats, are low, indicating that the model fails to recognize most of these abnormal heartbeats.

To improve the performance of the model on minority classes, we augment the original training set by adding synthetic data generated by our BioSGAN model to each minority class. We construct the augmented training set to have a more balanced sampling ratio of approximately $(5:1:1:1:1)$. The performance of the CNN model is then evaluated on the augmented testing set.

Table II shows that the recall rates and F1-scores for the minority classes have significantly improved after adding the synthetic data to the training set. This experiment demonstrates the predictive power of the synthetic data generated by our BioSGAN model. Synthetic data can be used as a substitute for real data in classification tasks and can be effective in improving the performance of a deep learning model on imbalanced datasets.

V. Conclusion

In this study, we proposed a transformer-based label-guided GAN model for generating multi-dimensional time-series data of arbitrary length. The model employs label information as a conditional restriction to guide the GAN model to generate time-series data of multiple classes, leveraging transfer learning effects from the different classes. We used various visual comparison methods to demonstrate the similarity between the original data and the synthetic data, and quantitative metrics to compare with other existing time-series GANs. Our proposed

method showed a significant advantage compared to existing methods.

REFERENCES

[1] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces," *Journal of neural engineering*, vol. 15, no. 5, p. 056013, 2018. 1

[2] X. Li and V. Metsis, "Spp-eegnet: An input-agnostic self-supervised eeg representation model for inter-dataset transfer learning," in *International Conference on Computing and Information Technology*. Springer, 2022, pp. 173–182. 1

[3] D. Ramyachitra and P. Manikandan, "Imbalanced dataset classification and solutions: a review," *International Journal of Computing and Business Research (IJCBR)*, vol. 5, no. 4, pp. 1–29, 2014. 1

[4] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019. 1

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014. 1, 2

[6] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690. 1

[7] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3722–3731. 1

[8] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5907–5915. 1

[9] E. Brophy, Z. Wang, Q. She, and T. Ward, "Generative adversarial networks in time series: A survey and taxonomy," *arXiv preprint arXiv:2107.11098*, 2021. 1

[10] O. Mogren, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv:1611.09904*, 2016. 2

[11] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional gans," *arXiv preprint arXiv:1706.02633*, 2017. 2

[12] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," *Advances in neural information processing systems*, vol. 32, 2019. 2, 3, 6

[13] H. Ni, L. Szpruch, M. Wiese, S. Liao, and B. Xiao, "Conditional sig-wasserstein gans for time series generation," *arXiv preprint arXiv:2006.05421*, 2020. 2, 3, 6

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008. 2, 3

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. 2, 4

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. 2

[17] Z. Wang, C. Zhao, and Z. Ni, "Adversarial vision transformer for medical image semantic segmentation with limited annotations," 2022. 2

[18] Z. Wang, T. Li, J. Zheng, and B. Huang, "When cnn meet with vit: Towards semi-supervised learning for multi-class medical image semantic segmentation. incomputer vision–eccv 2022 workshops: Tel aviv, israel, october 23–27, 2022," *Proceedings, Part VII*, pp. 424–441, 2023. 2

[19] Z. Wang, J.-Q. Zheng, and I. Voiculescu, "An uncertainty-aware transformer for mri cardiac semantic segmentation via mean teachers," in *Medical Image Understanding and Analysis: 26th Annual Conference, MIUA 2022, Cambridge, UK, July 27–29, 2022, Proceedings*. Springer, 2022, pp. 494–507. 2

[20] K. Lu, A. Grover, P. Abbeel, and I. Mordatch, "Pretrained transformers as universal computation engines," *arXiv preprint arXiv:2103.05247*, 2021. 2

[21] Y. Jiang, S. Chang, and Z. Wang, "Transgan: Two pure transformers can make one strong gan, and that can scale up," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 2

[22] S. Diao, X. Shen, K. Shum, Y. Song, and T. Zhang, "Tilgan: Transformer-based implicit latent gan for diverse and coherent text generation," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 4844–4858. 2

[23] X. Li, V. Metsis, H. Wang, and A. H. H. Ngu, "Tts-gan: A transformer-based time-series generative adversarial network," *arXiv preprint arXiv:2202.02691*, 2022. 2, 3

[24] A. Grinsted, J. C. Moore, and S. Jevrejeva, "Application of the cross wavelet transform and wavelet coherence to geophysical time series," *Nonlinear processes in geophysics*, vol. 11, no. 5/6, pp. 561–566, 2004. 2, 3, 5, 9

[25] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014. 2

[26] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410. 2

[27] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232. 3

[28] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *International conference on machine learning*. PMLR, 2017, pp. 2642–2651. 3

[29] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797. 3

[30] C. Cassisi, P. Montalto, M. Aliotta, A. Cannata, and A. Pulvirenti, "Similarity measures and dimensionality reduction techniques for time series data mining," *Advances in data mining knowledge discovery and applications'(InTech, Rijeka, Croatia, 2012,*, pp. 71–96, 2012. 3

[31] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223. 4

[32] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017. 4

[33] D. Micucci, M. Mobilio, and P. Napoletano, "Unimib shar: A dataset for human activity recognition using acceleration data from smartphones," *Applied Sciences*, vol. 7, no. 10, 2017. 5

[34] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001. 5

[35] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000. 5

[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. 5

[37] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012. 9

[38] Mathworks. Wavelet coherence matlab tutorial. [Online]. Available: https://www.mathworks.com/help/wavelet/ug/compare-time-frequency-content-in-signals-with-wavelet-coherence.html 10

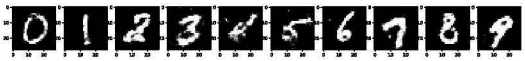
[39] X. Cui, D. M. Bryant, and A. L. Reiss, "Nirs-based hyperscanning reveals increased interpersonal coherence in superior frontal cortex during cooperation," *Neuroimage*, vol. 59, no. 3, pp. 2430–2437, 2012. 10

APPENDIX

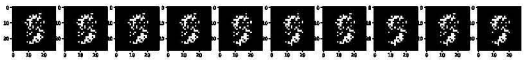
A. A Study of Conditional GAN Label Embedding Strategies

Since synthetic time-series data are hard to interpret, we can not easily know which label embedding strategy will perform best when building a conditional GAN model. Therefore, we choose a simple machine learning dataset, MNIST [37], trying to help us derive the best label embedding strategy. To mimic the signal-generating process, we flatten the MNIST digit images to 1D vectors. These vectors can be viewed as sequential inputs to the GAN models. We test a combination of label embedding methods to generate vectorized digit images. Then we reshape these vectors to the original image size and evaluate their fidelity. We listed a few experimental results of generated digit images with different label embedding strategies in Fig. 6.

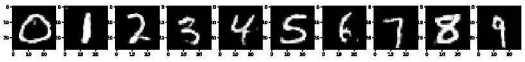
(a) Image samples from Real MNIST datasets



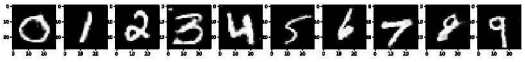
(b) Strategy 1



(c) Strategy 2



(d) Strategy 3



(e) Strategy 4 Our method

Fig. 6: Conditional Label Embedding strategies.

Strategy 1: Concatenate label embeddings to both Generator and Discriminator inputs on a linear-GAN. In this experiment, we build a GAN model in which both the generator and discriminator only contain a few fully-connected layers. In this strategy, the embedded label vectors are concatenated to the end of the input vectors. For example, if the generator input is a 1D vector of size $(100, 1)$ and the embedded label is a 1D vector of size $(10, 1)$. The label-embedded input to the generator will be a 1D vector of size $(110, 1)$. The plots in Fig. 6b show that this strategy generated digit images that are recognizable but contain a lot of noise.

Strategy 2: Add label embeddings to both Generator and Discriminator inputs on linear-GAN. In this experiment, we use the same linear GAN model as strategy 1. But instead of concatenating label embedding to the input data, we sum their values together. For example, if the generator input and embedded label are both 1D vectors of size $(100, 1)$, we add these two vectors together to get another vector that also has the size $(100, 1)$. The idea is similar to the strategy commonly used by Transformer models for embedding positional encoding to the input tokens. However, from the images in 6c, we can see such a strategy does not work well for label embedding.

Strategy 3: Concatenate label embeddings to both Generator and Discriminator inputs on a Convolutional Conditional GAN (CNN-GAN). In this experiment, we build a Conditional GAN with convolution layers to generate categorical digit images. The label embedding information is concatenated to the image feature maps as a separate channel. For example, after the Convolution layer transformation, the input image vector is reshaped to a feature map with the size $(Channel, height, width)$. The label embedding information is also shaped by a linear layer with the size $(1, height, width)$. Then, we concatenate them to a feature map with the size $(Channel + 1, height, width)$. The images in 6d show that this strategy generates an acceptable quality of synthetic digit images.

Strategy 4 (Our method) Concatenate label embedding only to the Generator and add a classification head on the Discriminator. We use the same concatenation method and CNN-GAN model architecture as in strategy 3. But instead of adding label embedding information to both the generator and discriminator, we only these information to the generator and let the discriminator classify the input categories. The images in 6e show that this strategy produces the best generated image quality over all the other strategies.

Besides the strategies mentioned above, we also tested many other label embedding combinations and GAN model architectures. We summarize the experimental findings as follows:

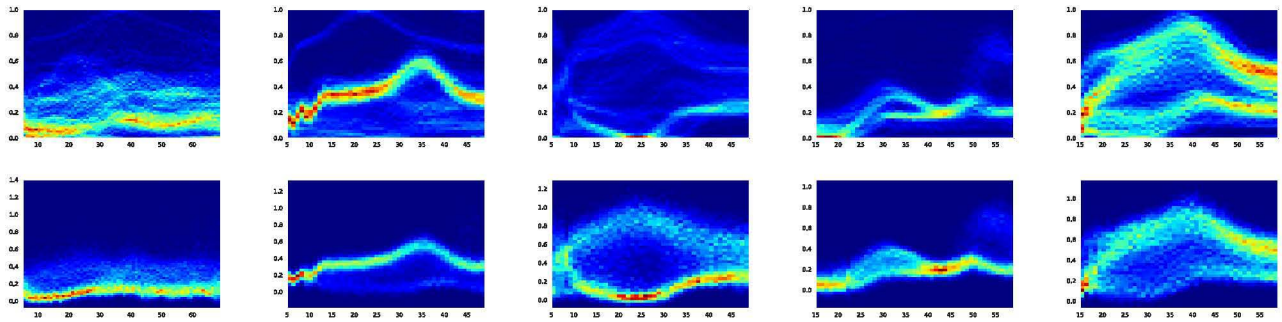
1) Concatenating label embedding to both the generator and discriminator inputs is a naive way to build a conditional GAN model. It may work but does not perform equally well on different model architectures.
2) Adding label embedding instead of concatenating, as in the transformer positional encoding, is not a practical solution for label embedding.
3) The minimum distortion rule applies. The label embedding should be concatenated with the input data with as little information as possible.
4) The more complex the GAN network, the better synthetic data ability it will generate.
5) Concatenating label embedding to the generator and adding a category classification head to the discriminator overall performed best on all kinds of GAN model architectures that we tested.

B. Wavelet Coherence Similarity Metric

Wavelet coherence ($wcoh$) [24] measures the correlation between two signals. The $wcoh$ of two time-series vectors x and y is:

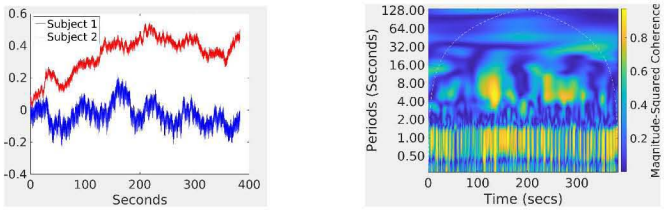
$$wcoh = \frac{|S(C_x^*(a,b)C_y(a,b))|^2}{S(|C_x(a,b)|^2) \cdot S(|C_y(a,b)|^2)} \tag{9}$$

where $C_x(a,b)$ and $C_y(a,b)$ denote the continuous wavelet transforms of x and y at scales a and position b. The



(a) (b) (c) (d) (e)

Fig. 7: MIT-BIH Arrhythmia ECG dataset real and synthetic signals fusion heatmaps. The top row images are from real signals, and the bottom row images are from synthetic signals. x-axis: signal timesteps; y-axis: signal microvolt (mV) value at each timestep. (a) Non-Ectopic Beats, (b) Superventrical Ectopic, (c) Ventricular Beats, (d) Unknown, (e) Fusion Beats.



(a) NIRS raw signals.

(b) NIRS Wavelet Coherence plot.

Fig. 8: NIRS signals and their corresponding wavelet coherence plot.

superscript $*$ is the complex conjugate and S is a smoothing operator in time and scale. Wavelet coherence is useful for analyzing non-stationary[2] signals. The inputs x and y must be of equal in length, one-dimensional real-valued signals. The coherence is computed using the analytic Morlet wavelet.

We use an example of finding coherent oscillations in human brain activities to better illustrate the properties of the Wavelet Coherence ($wcoh$) metric [38]. Fig. 8a shows two human subjects' brain activities recorded as near-infrared spectroscopy (NIRS) data [39]. When recording the data, the subjects are cooperating on a task repeated periodically for approximately 7.5 seconds. However, by looking at these two raw signals, we can hardly tell any similarities between them. When using the $wcoh$ metric to compute the correlation between these two signals, we can easily observe some common behaviors that co-exist between these two subjects. Fig. 8b shows a wavelet coherence plot of the two signals plotted in Fig. 8a. In Fig. 8b, the x-axis shows the length of these two signals, which is around 400 seconds. The $wcoh$ computation process will decompose a non-stationary sequence into many stationary sequences, and each stationary sequence has a constant frequency and phase. Therefore, if we look at the y-axis of this wavelet coherence plot, we can see that these two signals are decomposed into many sequences with multiple periods[3]. We can see that in period 1, these two signals show a high correlation over the whole 400 seconds. It represents the cardiac rhythms of the two subjects, which are commonly similar among human beings. In addition, we can observe a strong correlation at around the period of 7-8 within 100-300 seconds. It represents the two subjects periodically doing the same task every 7.5 seconds.

C. Visualizations with fusion maps

In Fig. 7, we plot five different categories of real and synthetic Arrhythmia ECG signals to fusion maps. Each fusion map is a combination of 1000 raw signals from a data category. We segment around 35-50 timesteps that are most meaningful for diseases diagnosing from each signal and draw the fusion maps. Those maps are represented as heatmaps where the brighter color indicates more data points having similar values. From these plots, we can see a similar data value distribution of real signals and synthetic signals. Since the 1000 signal samples we used to draw the fusion maps are randomly selected from the real and synthetic sets, their signal value distribution may be slightly different. For example, (a) and (c) real and synthetic fusion maps look a little different.

This fusion map strategy is a more direct way to observe data points' value distribution from multiple signals. It works well for ECG signals, but it may fail to picture such similarities when a category of data has multiple dimensions and has various signal patterns, such as the signals from the UniMiB datasets.

[2]A signal is said to be non-stationary if its frequency of spectral contents is changing with respect to time.

[3]The period is inversely proportional to frequency.